Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1-16. (Canceled)

17. (Currently amended) ~~Method~~ A computer implemented method for dynamically assembling a topology tree of a ~~Portal~~ portal page ~~by means of a Portal – Server application~~, comprising ~~the steps of~~:

loading an existing topology tree of a portal page from storage, wherein the existing topology tree comprises a plurality of nodes;

marking a node [[nodes]] of [[an]] the plurality of nodes of the existing topology tree ~~of a certain Portal page as~~ a dynamic node ~~Dynamic Nodes~~, wherein said ~~Dynamic Node~~ dynamic node comprises a node of said plurality of nodes that references a fragment placeholder, and wherein said fragment placeholder comprises a node that is to be resolved at runtime;

~~calling~~ assigning, by a dynamic assembly module, an ~~Adapter being assigned~~ adapter to said dynamic node ~~Dynamic Node by means of said Dynamic Assembly Module~~;

creating, using said adapter, ~~an appropriate~~ a topology tree fragment, wherein said topology tree fragment comprises a partial topology tree that is built ~~by means of said Adapter~~ at runtime; and

replacing, by said dynamic assembly module, said fragment placeholder [[by]] with said topology tree fragment ~~by means of said Dynamic Assembly Module~~ at runtime.

18. (Currently amended) ~~Method~~ The computer implemented method according to claim 17, further comprising the [[steps]] step of:

assigning an expiry time to the topology fragment, wherein the topology fragment does ~~fragments which will~~ not change for a given period of time~~; and~~, wherein replacing, by said dynamic assembly module, said fragment placeholder [[by]] with said topology fragment ~~by means of said Dynamic Assembly Module~~ at runtime, comprises:

replacing, by said dynamic assembly module, said fragment placeholder at runtime if said expiry time is not expired.

19. (Currently amended) ~~Method~~ The computer implemented method according to [[said]] claim 17, further comprising the [[steps]] step of:

caching said topology fragment, wherein ~~fragments; and~~ replacing said fragment placeholder at runtime comprises:

replacing, by said dynamic assembly module, said fragment placeholder by said cached topology fragment ~~fragments by means of said Dynamic Assembly Module~~ at runtime if said adapter ~~Adapter~~ does not receive a notification ~~from said External System~~ that said topology fragment has been changed.

20. (Currently amended) ~~Method~~ The computer implemented method according to claim 17, wherein said ~~Adapter~~ adapter is a workflow adapter, and further ~~Workflow Adapter, wherein said Workflow Adapter~~ comprising the steps of:

obtaining a workflow item from a workflow system ~~Workflow System~~;
mapping said workflow item to an associated workflow portlet ~~Workflow Portlet~~;
creating a topology fragment containing said workflow portlet ~~Portlet~~; and
returning said topology fragment to said dynamic assembly module ~~Dynamic Assembly Module~~.

21. (Currently amended) ~~Method~~ The computer implemented method according to claim 20, wherein said workflow portlet ~~Workflow Portlet~~ ~~provides means to perform the~~ performs tasks associated with said [[work]] workflow item.

22. (Currently amended) ~~Method~~ The computer implemented method according to claim 21, wherein said ~~Workflow Portlet~~ workflow portlet calls the workflow system ~~Workflow System~~ to check out the [[work]] workflow item and to check ~~checks~~ in the [[work]] workflow item.

23. (Currently amended) ~~Method~~ The computer implemented method according to claim 22, wherein said ~~Workflow Portlet~~ workflow portlet receives said [[work]] workflow item from a workflow system service ~~Workflow System~~ ~~Service~~.

24. (Currently amended) ~~Method~~ The computer implemented method according to claim 23, wherein said ~~Workflow System Service~~ workflow system service communicates with said workflow system ~~Workflow System~~ via a ~~Workflow System~~ workflow system API which obtains the work items of a user, checks out said workflow item ~~work items~~, and checks in said [[work]] workflow item.

25. (Canceled)